

REMARKS

After entry of the above amendments, claims 21-36 are pending in this application. Applicants have canceled claims 1-20 and added new claims 21-36. No new matter has been added. New claim 21 incorporates the limitations of previously presented claims 1, 3 and 4. New claim 27 incorporates the limitations of previously presented claims 9, 12 and 15. The remaining dependent claims correspond to previously presented dependent claims.

Objections to the Specification

Applicants have amended the specification and Abstract. No new matter has been added.

Rejections Under 35 U.S.C. § 102 and 35 U.S.C. § 103

Claims 1-3, 5, 6, 9-14, 16, 17 and 19 stand rejected under 35 U.S.C. 102(b) as anticipated by Barker U.S. Patent No. 5,353,421. Claims 4, 7, 8, 15, 18 and 20 stand rejected under 35 U.S.C. 103(a) as unpatentable over Barker, Panikatt U.S. Patent No. 6,349,333 and Kampe U.S. Patent Pub. No. 2002/0016867. Applicants respectfully traverse these rejections.

Applicants' claims are directed to a method and system for managing and transmitting events from a server to a client in which the client sees the data transmission being initiated by the server. In the method, for every event that needs to be transmitted from a server via a communication link to a client, for example a client application, the event is logged using a client event service and a server event service. Only events for which logging has been performed are transmitted from the

server to the client. Such event logging prompts a respective update, or the first logging prompts an initialization, of the client/server system.

When an event occurs, it is first reported to an installation interface of the server. If the event in question has been logged, it is transferred from the installation interface to the server event service. The client event service uses the communication link to make requests for event transmission to the server event service, for example periodically. If there is an event that has been detected by the server event service, it is transmitted via the communication link to the client event service based on the received request.

Within the client, the client event service transmits received events to the client application, where the event is reported, for example, by producing an entry that describes the event in an event list. Transmitting an event that has occurred to the client application therefore requires no active requests from the client application. Since the client application does not communicate with the server, but rather only with the client event service, it is independent of the server. When the method is used, the client application sees the event handling operation taking place as in a local environment.

In one embodiment, the client application logs an appropriate client callback function in the client event service for every event about which it is to be notified. The client event service then uses the communication link to log a corresponding server callback function in the server event service. This logging is carried out separately for every event about which the client application is to be notified. In this way, it is possible to handle all events independently of one another. An association can be made, for example in the form of a list, in preparation for the method, so that this association is used to assign a unique name to every event possibly occurring in

the installation. This association exists in the client and in the server. Thus, every event has the same associated name in the client and in the server. To log the callback functions, the client application calls a client logging function from the client event service and provides it with the name of the event in question and with a pointer to the client callback function that is to be logged.

The client logging function then generates a unique event identifier and transmits this event identifier together with the event name to a server logging function of the server event service via the communication link.

The server logging function logs a server callback function with the installation interface by transferring the event name. The server logging function then stores a data record, which contains the event identifier, a pointer to the server callback function which is to be logged and possibly further data, such as the event name, in a server event table. The server logging function then uses the communication link to report back to the client logging function of the client event service that the logging operation has been performed. The client logging function then logs the client callback function by storing a data record, which contains the event identifier, a pointer to the client callback function that is to be logged and possibly further data, such as the event name, in a client event table.

As the cited references were applied to the previously presented claims, Applicants respectfully submit that new independent claims 21 and 27 include a combination of elements that is neither disclosed nor suggested by Barker or the other references, viewed alone or in combination.

For example, Barker discloses a method for remotely managing a plurality of network elements of a telecommunications network through a special communication link, including a computer internet. However, Barker provides no teaching or

suggestion where the client event service transmits received events to the client application and where the event is reported, for example, by producing an entry that describes the event in an event list. As previously mentioned, transmitting an event that has occurred to the client application requires no active requests from the client application. Since the client application does not communicate with the server, but rather only with the client event service, it is independent of the server. The cited section in Barker (col. 4, lines 19-36) disclose that clients' commands generate HTTP requests to the element management system server and the server gathers the information, dynamically generates a web page and sends the results/output to the web browser for display.

Barker similarly does not disclose where requests initiated by the client event service regarding the detected events are made to the server event service, as recited in the claims. The cited section, Barker (col. 11, lines 21-28), specifies that it is the clients that register a filter with the Event Distributor in the object server to request delivery of events matching the filter. As previously mentioned, Applicants' client application does not communicate with the server, but rather only with the client event service.

Furthermore, neither Panikatt nor Kampe make up for the deficiencies in Barker. Accordingly, independent claims 21 and 27 are patentable, as are the remaining claims, which depend from claims 21 and 27.

Conclusion

For the foregoing reasons, Applicants respectfully submit that this application is in immediate condition for allowance and all pending claims are patentably distinct

from the cited references. Reconsideration and allowance of all pending claims are respectfully requested.

In the event that there are any questions about this application, the Examiner is requested to telephone Applicants' undersigned representative so that prosecution of the application may be expedited.

If additional fees are required for any reason, please charge Deposit Account No. 02-4800 the necessary amount.

Respectfully submitted,

BUCHANAN INGERSOLL & ROONEY PC

Date: July 11, 2008

By: /Brian N. Fletcher/
Brian N. Fletcher
Registration No. 51683

P.O. Box 1404
Alexandria, VA 22313-1404
703 836 6620